**Test Plan for Paint Calculator**

**Objective:**

Validate functionality of a paint calculator website both manually and programatically using automated tests.

**Scope:**

Paint calculator website features and all of its functional workflows will be validated.

**Entry / Exit criteria:**

**Entry criteria** - website is available for testing, test plan/ test cases are reviewed and ready to execute

**Exit criteria** - Requirements for the website are met, all test cases are executed, all high and medium priority defects are fixed, automation completed for applicable features

**Environment:**

Website running locally <http://127.0.0.1:5000/>

Test environment url - https://<test>:5000

Staging environment - https://<stage>:5000

**Assumptions:**

Test environment and website are available at least 90% of the time except for scheduled bug fixes and feature enhancements.

**Testing Types:**

The followingare the testing types that will be performed.

**BVTs - Build verification** -

Validates that the website is up and basic functionality works

**Functional testing** -

End user scenarios both happy path and unhappy path tests will be covered

**Performance testing** -

Validates the response time of the website, validates to see if there are any spikes in the CPU and memory on increasing the user load.

**Security testing** -

Tools such as appscan will be used to test the site for any security vulnerabilities such as cross site scripting, SQL injection etc. Website is validated with Javascript turned off, using tools such as firefox tamper data.

**Load testing** -

Validates how the website performs under load. System will be tested under short haul (5 hours) 10 user constant load as well as long haul (24 hours) 10 user constant load to monitor CPU usage, memory etc

**Stress testing -**

Validates how website performs on increasing load. System will be tested under a step up load on a short haul (5 hours) and then a long haul (24 hours) to monitor CPU usage, memory etc

**Failover testing** as applicable -

Assuming there is a disaster recovery site, validate that the DR site is accessible and functions the same as the primary site.

**User acceptance testing** -

Actual software users will be invited to exercise real world scenarios on test environment. This will help find any usability defects that can be fixed.

**Test strategy:**

Build verification tests and functional tests will be automated using C# in Visual studio. Performance tests will be executed using Visual studio. Security tests will be performed using Appscan. Performance, Stress and Load testing will be performed using Visual studio.

**Test scenarios:**

**Happy path tests** -

* Website is validated on multiple browsers - Chrome, IE, Edge, Firefox
* Website is validated for functionality and responsiveness on mobile device browsers - Android / iOS tablet, Android/ iOS phones - spec based on requirements, can use Sauce labs
* End to end workflows using 1+ rooms and co-ordinates adding up to > 400 sq.ft will be validated
* End to end workflows using 1+ rooms and co-ordinates adding up to < 400 sq. ft will be validated
* Validate all buttons, links, navigations

**Unhappy path tests -**

* Validation of all fields for length checks, alpha characters, alphanumeric chars, Unicode chars, validate error handling
* Validation of website without Javascript and sending values using firefox tamper data to see error handling
* Enter 100+ rooms, validate end to end results calculation
* Boundary checks - enter 1000+ for number of rooms and validate website functionality

**Metrics:**

The following metrics will be computed

* Test case execution metrics
* Defect metrics
* Test case Pass/ fail metrics

**Risks:**

Upcoming holiday season (assuming), PTO can push test and release cycle by a few weeks.

Availability of test environment with test build deployed on <Date> and staging environment with stage build with bug fixes deployed on <Date> is necessary to meet Production date of <Date>

**Resources:**

Test team - Shalini Ramachandran and tester1

Dev team - dev1 and dev2

**Dependencies:**

Any relevant dependencies can be called out.